

ÜNİTE VII
ARİTMETİK DEVRELER

Binary sayılarla toplama, çıkarma, çarpma ve bölme gibi aritmetik fonksiyonları yapan devrelerdir. Sayısal bilgisayarlarda ve hesap makinelerinde esas işlemler toplama ve çıkarma işlemleridir. Çarpma işlemi, tekrarlanan toplama işlemleri ile bölme işlemi ise tekrarlanan çıkarma işlemleri ile yapılır.

A- Toplayıcılar

Lojik devrelerde kullanılan iki temel toplayıcı tipi vardır. İki bitin toplanmasını yapan devreye YARIM TOPLAYICI, üç bitin toplanmasını yapan devreye de TAM TOPLAYICI devresi adı verilir.

Toplayıcı devrelerine geçmeden önce basit binary toplama kuralları tekrar ele alalım:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10 \text{ (Sum = 0, carry = 1)}$$

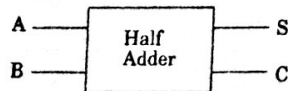
1- Yarım Toplayıcı (Half Adder)

Bu devre giriş değişkenlerini toplar ve toplanan bitleri ve çıkış değişkenlerini ise toplam (SUM) ve elde (CARY) oluşturur. Yarım toplayıcı devresi, en basit bir toplama devresidir. Burada yarım toplayıcı devresini tasarım yolu ile gerçekleştireceğiz.

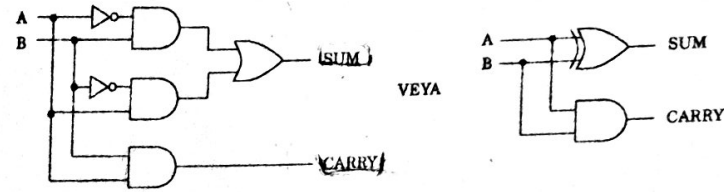
GİRİŞLER		ÇIKIŞLAR	
A	B	SUM (S)	CARRY (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\text{SUM (TOPLAM)} = \bar{A}B + A\bar{B} = A \oplus B$$

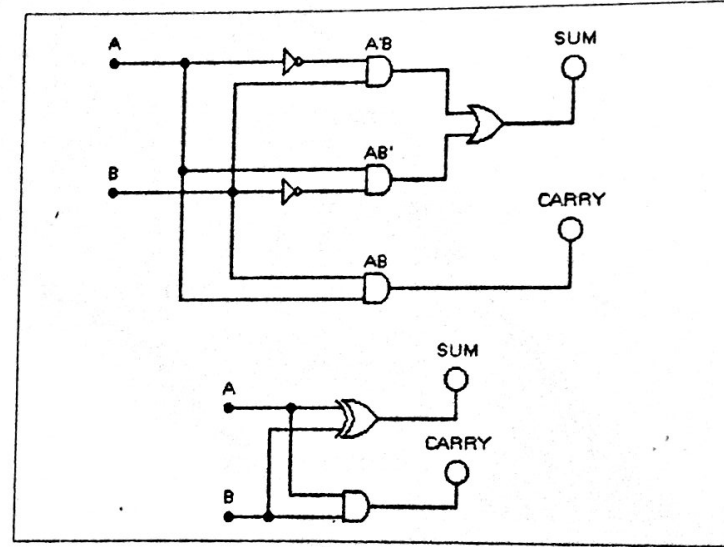
$$\text{CARRY (ELDE)} = AB$$



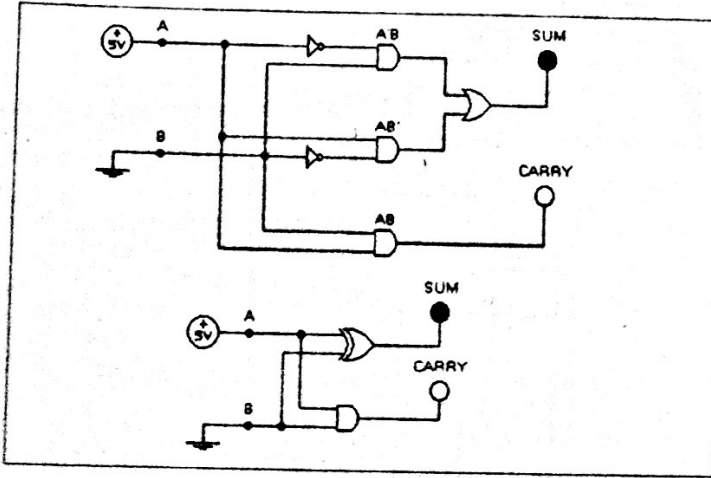
Tablo 7.1: Yarım toplayıcının doğruluk tablosu



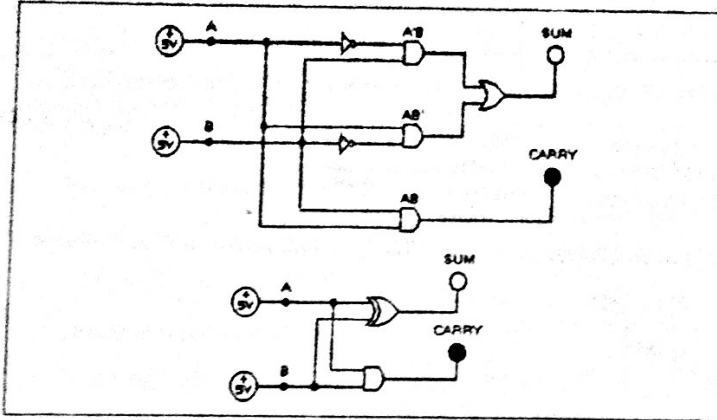
Şekil 7.1: Yarım Toplayıcı Devreleri



Şekil 7.2: Yarım toplayıcı devresinin Electronics Workbench programında uygulanmış şekli



Şekil 7.3: A=1 ve B=0 konumunda yarı toplayıcı devresinin çıkışından SUM=1 ve CARRY = 0 alınması



Şekil 7.4: A=B=1 konumunda SUM=0 CARRY=1 çıkışının elde edilmesi

Yarı toplayıcı (Half Adder) devresinde programda yapılan örneklerle göre; A=1 ve B=0 uygulandığında SUM çıkışının 1, CARRY çıkışının 0 olduğu, A=B=1 uygulandığında SUM çıkışının 0, CARRY çıkışının 1 olduğu doğrulanmıştır.

2- Tam Toplayıcı (Full Adder)

Girişindeki 3 bitin toplamını gerçekleştiren devredir. Bu devrenin 3 girişi ve 2 çıkışı olup, girişlerden ilk ikisi toplanacak iki değerlikli bit, son giriş ise bir önceki düşük değerlikli bitlerin toplamından gelen eldeyi gösterir. Yani girişlere A, B, C dersek, A ve B girişleri toplanacak iki biti gösterir. C giriş değişkeni ise, A ve B bitlerinin toplamındaki eldeyi (carry) varsa gösterir. Bir tam toplayıcı devresi iki yarı toplayıcının birleşiminden oluşur. Burada tam toplayıcı devresini iki türlü gerçekleştirebiliriz.

a) Tasarım Yoluyla Tam Toplayıcı Devresinin Gerçekleşmesi

GİRİŞLER	ÇIKIŞLAR
A B C	SUM CARRY
0 0 0	0 0
0 0 1	1 0
0 1 0	1 0
0 1 1	0 1
1 0 0	1 0
1 0 1	0 1
1 1 0	0 1
1 1 1	1 1

$$\text{SUM (TOPLAM)} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$\text{CARRY (ELDE)} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

Tablo 7.2: Tam toplayıcının doğruluk tablosu

Elde edilen çıkış ifadelerinin sadeleşip sadeleşmediğine bakılmalıdır. Eğer bu ifadelerin, en sade hali bulunmazsa tasarımı yapılacak tam toplayıcı devresi daha karmaşık olacak ve kapı adedi artacağı için maliyeti artacaktır. Bulunan çıkış ifadeleri 3 değişkenli olduğu için bu ifadeleri sadeleştirmek için 3 değişkenli Karnaugh diyagramı kullanılmalıdır.

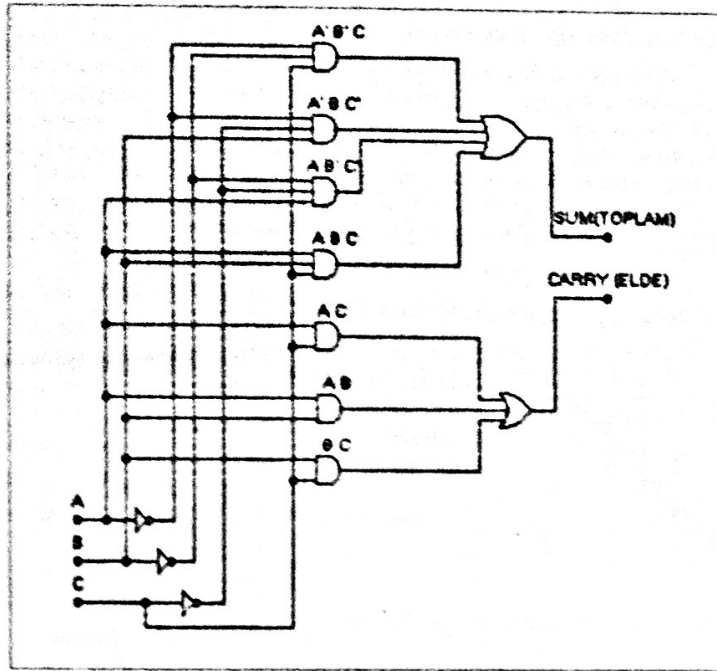
BC	00	01	11	10
A				
0		1		1
1	1		1	

$$S = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

BC	00	01	11	10
A				
0		1		1
1	1	1	1	1

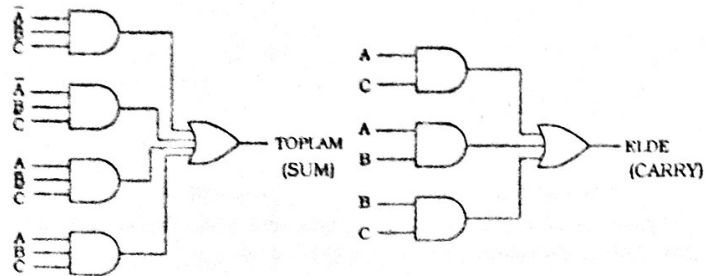
$$C = AC + AB + BC$$

Karnaugh diyagramından görüldüğü gibi SUM ifadesi sadeleşmezken, CARRY ifadesinin en sade hali elde edildi.

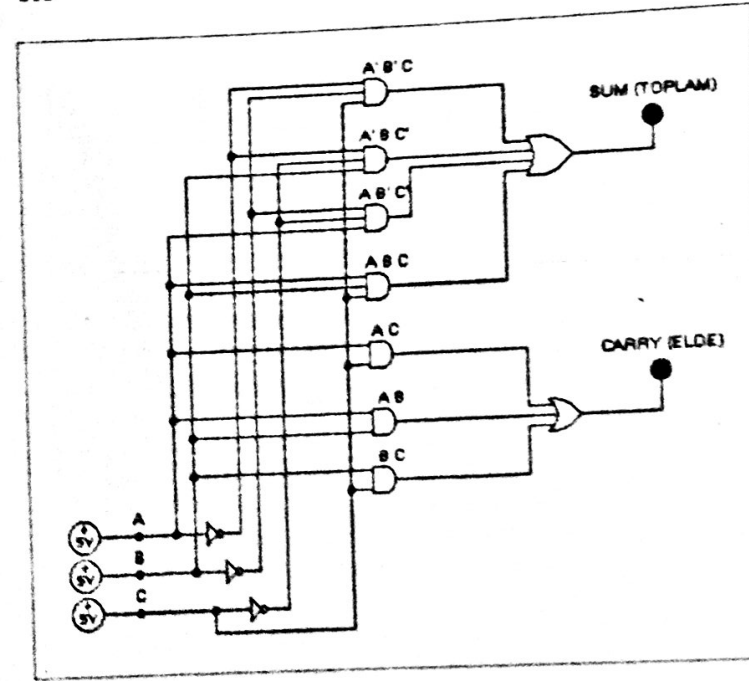


Şekil 7.5: Tam Toplayıcı Devresi

Şekil 7.5 teki tam toplayıcı devresinde girişlerin sadece kendileri uygulanmıştır. Girişlerin hem kendisinin hem de komplementlerinin (tersi) verildiği tam toplayıcı devresini aşağıdaki gibi göstermek mümkündür.



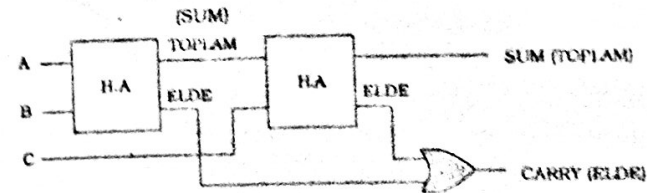
Şekil 7.6: Tam Toplayıcı Devresi



Şekil 7.7: Tam Toplayıcı Devresinin EWBD programında uygulanış şekli

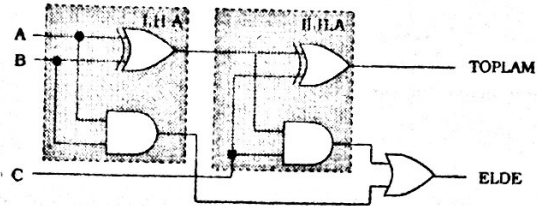
Tasarım yoluyla elde edilen tam toplayıcı devresinde yapılan örneklerde $A=B=C=1$ konumunda devrenin iki çıkışından 1 alınmış. $A=C=1$ ve $B=0$ konumunda ise sadece ELDE (CARRY) çıkışından 1 alınacaktır.

b) İki Half Adder'in Birleştirilmesiyle Full Adder'in Elde Edilmesi



Şekil 7.8: Tam Toplayıcı Devresi

Şekil 7.8'deki blok diyagramdan anlaşılacağı gibi I. Half Adder'ın TOPLAM çıkışı, II. Half Adder'ın girişine, C girişi de II. H. A.'ın diğer girişine uygulanır. İki H.A.'ın ELDE çıkışları ise bir OR kapısıyla birleştirilerek, tam toplayıcı devresinin ELDE çıkışını oluşturur. Yukarıdaki Half Adder paketlerini kaldırıp yerine açık devrelerini yerleştirirsek Şekil 7.9'daki gibi tam toplayıcı devresini bulmuş oluruz.



Şekil 7.9: Tam Toplayıcı Devresi

İki adet Half Adder'ın birleştirilmesiyle meydana gelen tam toplayıcı devresi Şekil 7.9'daki gibidir. Şimdi, bu çizilen devrenin gerçekten tam toplayıcı devresinin şekli olup olmadığını araştıralım. Yani bu devrenin tam toplayıcı devresinin şekli olduğunu ispatlayalım. Yukarıdaki TOPLAM ve ELDE çıkış ifadelerinin, tasarım yolu ile elde ettiğimiz çıkış ifadelerinin aynı olup olmadığını araştıracağız. Bu sayede Boolean Cebirini tekrar gözden geçirmiş olacağız:

$$\text{TOPLAM} = (A \oplus B) \oplus C$$

$$= X$$

$$= X \oplus C$$

$$= \bar{X}C + X\bar{C}$$

$$= (A \oplus B) \cdot C + (A \oplus B) \cdot \bar{C}$$

$$= (\bar{A}\bar{B} + AB) \cdot C + (\bar{A}B + A\bar{B}) \cdot \bar{C}$$

$$= \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

A \ BC	00	01	11	10
0		1		1
1	1		1	

$$\text{TOPLAM} = \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

$$\text{ELDE} = (A \oplus B) \cdot C + AB$$

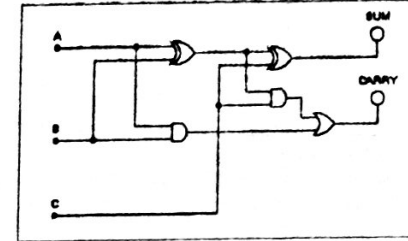
$$= (\bar{A}\bar{B} + A\bar{B}) \cdot C + AB$$

$$= \bar{A}\bar{B}C + A\bar{B}C + AB$$

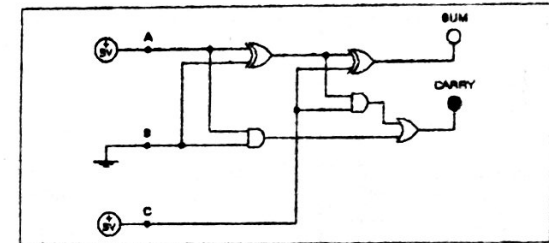
A \ BC	00	01	11	10
0			1	
1		1	1	1

$$\text{ELDE} = AC + AB + BC$$

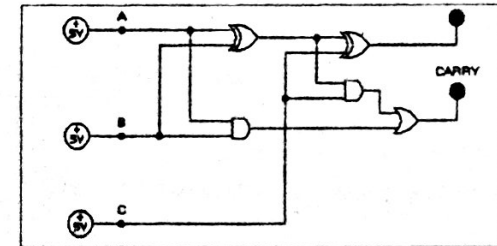
Bulduğumuz TOPLAM ve ELDE ifadeleri, tasarım yoluyla bulunan çıkış ifadelerinde aynı olduğundan, ikinci olarak çizilen tam toplayıcı devresinin doğru olduğu sonucuna varılır.



Şekil 7.10: Tam Toplayıcı devresinin Electronics Workbench programında uygulanmış şekli



Şekil 7.11: A=C=1 ve B=0 durumunda SUM çıkışının 0, CARRY çıkışının 1 olarak elde edilmesi

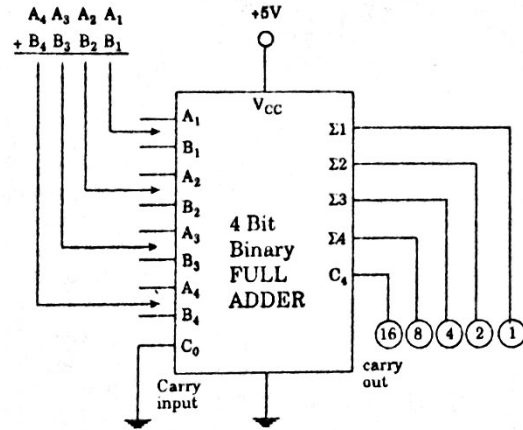


Şekil 7.12: A=B=C=1 konumunda SUM ve CARRY çıkışlarından 1 elde edilmesi

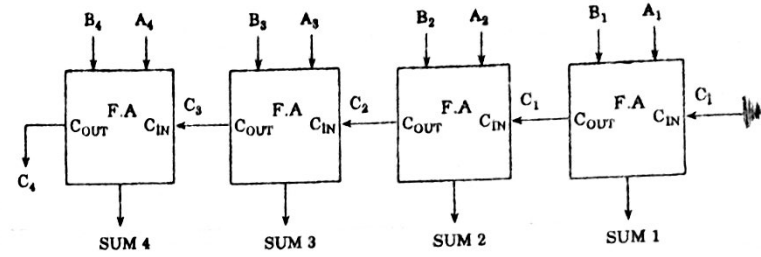
Tam toplayıcı devresini Electronics Workbench programında çalıştırdığımızda; $A=C=1$ ve $B=0$ konumunda SUM (Toplam) çıkışının 0, CARRY (elde) çıkışının 1 olduğu, $A=B=C=1$ konumunda SUM ve CARRY çıkışlarının 1 olduğu tespit edilmiştir. Diğer giriş kombinasyonlarında da bu devre çalıştırılabilir.

3) Paralel Toplayıcı

n bitlik iki binary sayıyı toplayan devreye paralel toplayıcı denir. Paralel toplayıcı devresinde n sayıda tam toplayıcı devresi kullanılır. Devreye A ve B binary sayılarının tüm bitleri aynı anda uygulanır. Tam toplayıcıdan gelen çıkış eldesi, bir sonraki (solundaki) tam toplayıcının giriş eldesine bağlanır. n bitlik paralel toplayıcılar, MSI entegrelerine bir örnektir. Paralel toplayıcılarda, full adder'lar paket olarak gösterilir. Şekil 7.13 ve 14'te 4 bitlik iki binary sayıyı toplayabilen paralel toplayıcı devresi gösterilmektedir. C_1 , paralel toplayıcının giriş eldesini, C_4 ise çıkış eldesini gösterir. SUM1, SUM2, SUM3 ve SUM4 çıkışları toplam bitleri gösterir. Örnek olarak 7483 IC tipli TTL entegreleri gösterilebilir.



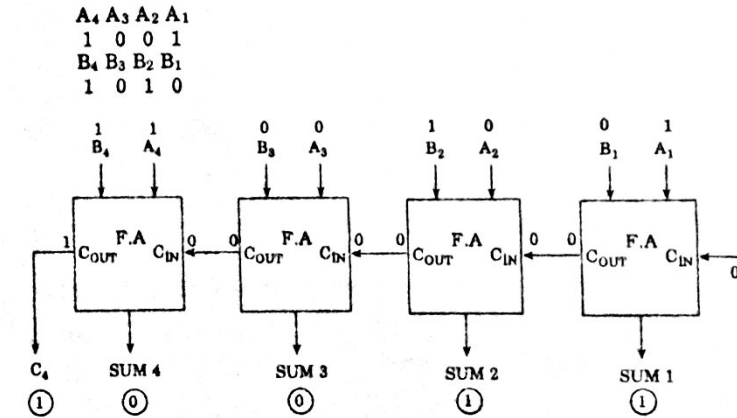
Şekil 7.13: 7483 Entegresiyle Yapılan Toplayıcı Devresi



Şekil 7.14: 4 Bitlik Paralel Toplayıcı Devresi

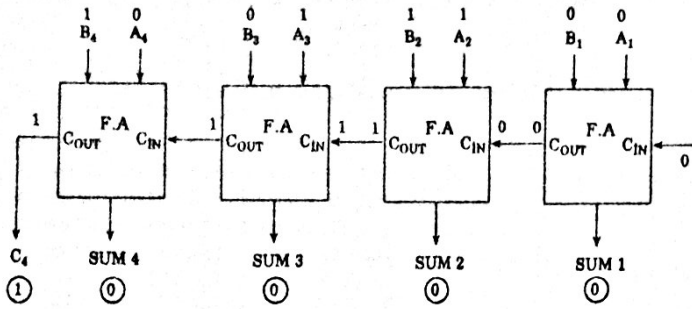
Örnek 1: 1001 ve 1010 sayılarını toplayabilen paralel toplayıcı devresini çiziniz ve işlemi devre üzerinde yapınız.

Verilen iki binary sayı 4 bitlik olduğu için, paralel toplayıcıda 4 adet paket (Full Adder) kullanılmalıdır. Verilen örnekte $A=1001$ $B=1010$ olarak verilmiştir.



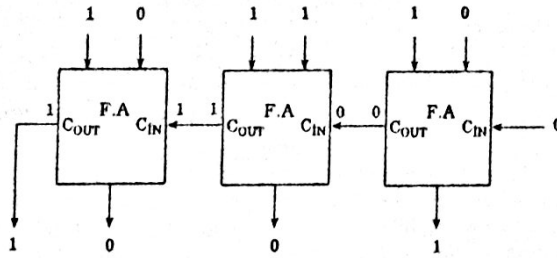
Şekil 7.16: 4 Bitlik Paralel Toplayıcı

Örnek 2: 4 bitlik iki sayıyı toplayabilen bir paralel toplayıcı devresinin blok diyagramını çiziniz ve iki örnek sayıyı (0110+1010) sayılarını toplayıp sonucu yazınız.



Şekil 7.16: 4 Bitlik Paralel Toplayıcı

Örnek 3: Üç bitlik paralel toplayıcı çizerek (2+7) işlemini yapınız.



Şekil 7.17: 3 bitlik paralel toplayıcı devresi

B) Çıkarıcı Devreler (Subtractor Circuits)

Lojik devrelerde kullanılan iki temel çıkarıcı tipi vardır. İki bitin çıkarmasını yapan devreye YARIM ÇIKARICI, üç bitin çıkarmasını yapan devreye de TAM ÇIKARICI devresi adı verilir.

1- Yarım Çıkarıcı (Half Subtractor):

GİRİŞLER		ÇIKIŞLAR	
A	B	Difference (FARK) A-B	BORROW (BORÇ)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Tablo 7.3: Yarım çıkarıcının doğruluk tablosu

Yarım çıkarıcı devresinde iki giriş ve iki çıkış bulunur. A-B olursa, üç durum vardır:

0-0 = 0
1-0 = 1
1-1 = 0

Bu sonuçlara FARK (DIFFERENCE) BİTİ denir. A<B olduğu zaman (0-1) olur ki, bir yüksek kademedeki 1 borç (BORROW) alınır. Yarım Çıkarıcı devresinde iki çıkış olup, çıkışlardan birisi

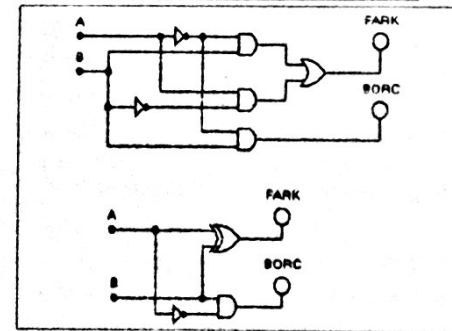
farkı üretir, diğer çıkış ise bir sonraki kademeye 1 borç verildiği bilgisini veren binary sayıyı üretir. Bu devrenin doğruluk tablosu basit çıkarma kurallarına göre düzenlenmiştir. Tekrar çıkarma kurallarını ele alalım.

0-0 = 0
1-1 = 0
1-0 = 1
0-1 = 1 (Borç = 1)

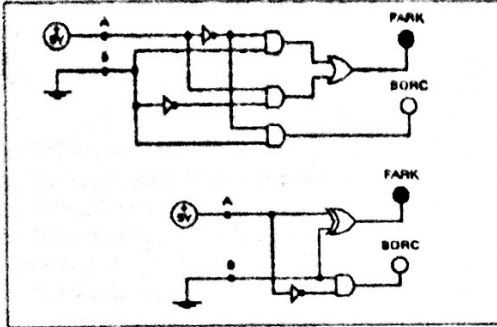
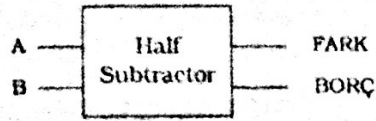
Yarım çıkarıcı devresinin, doğruluk tablosuna göre çıkış ifadeleri aşağıdaki gibi olacaktır:

$$D (\text{DIFFERENCE}) = \bar{A}B + A\bar{B} = A \oplus B$$

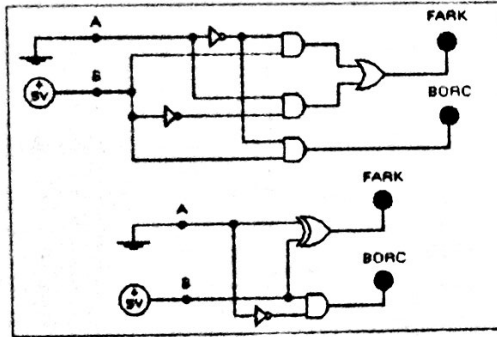
$$B (\text{BORROW}) = \bar{A}B$$



Şekil 7.18: Yarım çıkarıcı devresinin Electronics Workbench programında uygulanmış şekli



Şekil 7.19: Yarım çıkartıcı devresinin A girişine 1, B girişine 0 uygulandığı zaman ki durumu



Şekil 7.20: Yarım çıkartıcı devresinin A girişine 0, B girişine 1 uygulandığı zaman devrenin şekli

Yarım çıkartıcı devresine alt örneklerde A girişine 1, B girişine 0 uygulandığı zaman $FARK = 1$, $BORÇ = 0$; tam tersi durum uygulandığında ise $FARK$ ve $BORÇ$ çıkışlarının 1 olduğu gözlenmiştir.

2- Tam Çıkartıcı (Full Subtractor)

Daha düşük değerlikli kademeden 1 borç alınmış olabileceği dikkate alınarak iki bitü birbirinden çıkaran kombinasyonel bir devredir. Bu devre 3 girişe ve 2 çıkışa sahiptir. Girişlere A, B ve C dersek; A çıkarılan,

B çıkan, C ise borcu gösterir. Çıkışlardan biri farkı gösterirken, diğeri borcu gösterir.

a) Tasarım Yoluyla Tam Çıkartıcı Devresinin Elde Edilmesi

GİRİŞLER			ÇIKIŞLAR	
A	B	C	FARK	BORÇ
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Girişler 3 adet olduğu için alabileceği toplam değişik durum 8 olacaktır. $A=B=0$, $C=1$ durumunda, C biti daha düşük kademeden daha önce 1 borç alındığını gösterir. A-B işlemi gerçekleştirilerek daha sonra C biti ile gerçekleştirilir. $Fark = 1$, $Borç = 1$ bulunur. Devrenin çıkış ifadeleri;

$$FARK = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$BORÇ = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + ABC$$

Tablo 7.4: Tam çıkartıcının doğruluk tablosu

Bulunan bu çıkış ifadelerinin sadeleştirip sadeleşmediğine bakmak gerekir.

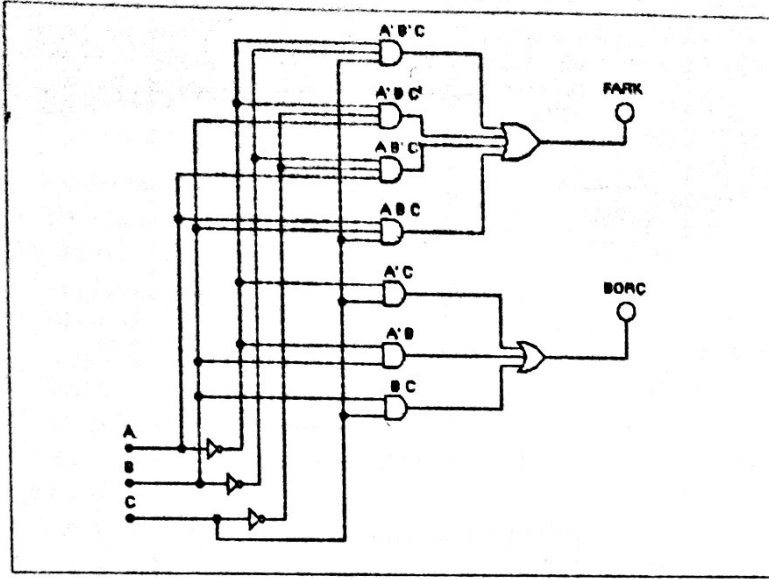
A	BC			
	00	01	11	10
0		1		1
1	1		1	

A	BC			
	00	01	11	10
0		1	1	
1	1		1	

$$FARK = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

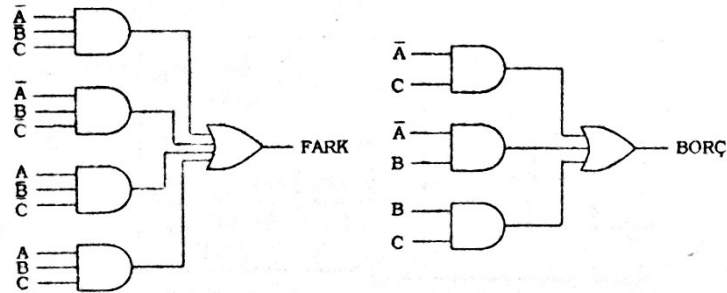
$$BORÇ = \bar{A}C + \bar{A}B + BC$$

Yapılan işlemden görüleceği gibi $FARK$ ifadesi sadeleşmeyip, $BORÇ$ ifadesinin en sade hali elde edilmektedir.

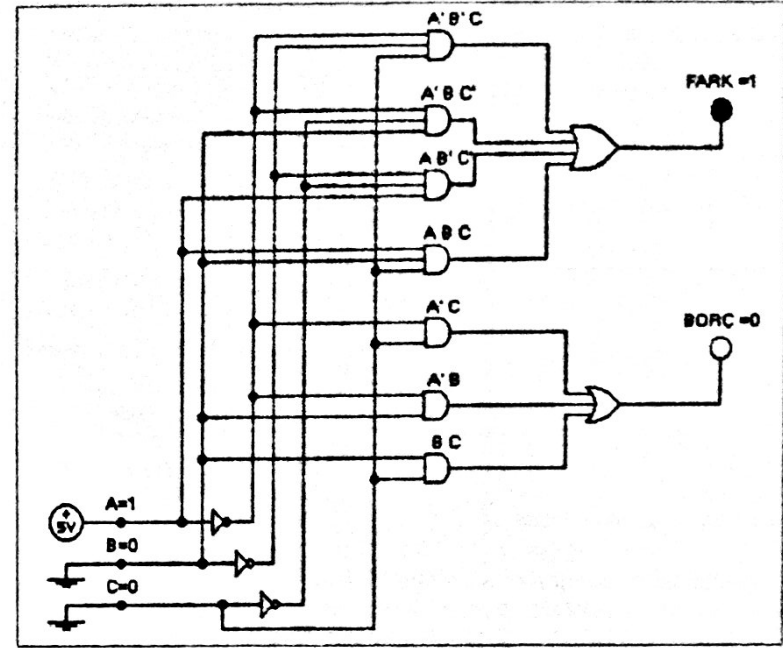


Şekil 7.21: Tam Çıkarıcı Devresi

Şekil 7.21'deki tam çıkarıcı devresinde girişlerin sadece kendileri uygulanmıştır. Girişlerin hem kendisinin hem de komplementlerinin (değil) verildiği tam çıkarıcı devresi Şekil 7.22'deki gibi olacaktır. Aşağıda çizilen lojik devrelerin çıkış ifadeleri çarpımların toplamı şeklindedir.



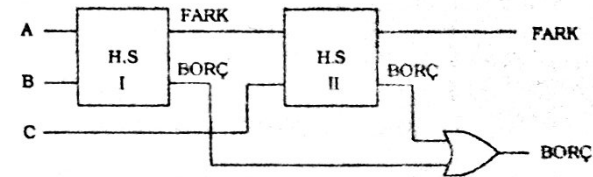
Şekil 7.22: Tam Çıkarıcı Devresi



Şekil 7.23: Tam çıkarıcı devresinin EWBD programında uygulanmış şekli

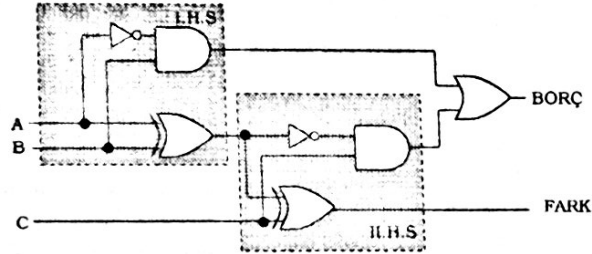
Tasarım yoluyla elde edilen tam çıkarıcı devresinde yapılan örneklerde, $A=1$ ve $B=C=0$ uygulandığında FARK çıkışının 1, BORÇ çıkışının 0 olduğu; $A=C=0$ ve $B=1$ konumunda ise her iki çıkışın da 1 olduğu saptanmıştır.

b) İki Yarım Çıkarıcının Birleştirilmesi ile Tam Çıkarıcının Elde Edilmesi:



Şekil 7.24: Tam Çıkarıcı Devresi

Şekil 7.24'deki blok diyagramdan anlaşılacağı gibi I. Half Subtractor'ın FARK çıkışı, II. Half Subtractor'ın girişine, C girişi II. Half Subtractor'ın diğer girişine uygulanır. İki Half Subtractor'ın BORÇ çıkışları bir OR gate ile birleştirilerek, tam çıkarıcı devresinin BORÇ çıkışı elde edilir. Şekil 6.85'deki half subtractor paketlerini kaldırıp, yerine açık şemalarını yerleştirirsek Şekil 7.25'deki tam çıkarıcı devresi elde edilir.



Şekil 7.25: Tam Çıkarıcı Devresi

Çizilen bu devrenin gerçekten tam çıkarıcı devresi olduğunu ispat edelim. Bu devrenin FARK ve BORÇ çıkışlarını bularak tasarım yoluyla bulduğumuz FARK ve BORÇ çıkışları ile aynı olup olmadığına bakalım.

$$FARK = C \oplus (A \oplus B)$$

$$\begin{aligned} A \oplus B &= X \\ &= C \oplus X \\ &= CX + C\bar{X} \\ &= \bar{C}(AB + A\bar{B}) + C(\bar{A}\bar{B} + AB) \\ &= \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + ABC \end{aligned}$$

$$\begin{aligned} BORÇ &= \bar{A}\bar{B} + (\bar{A} \oplus B) \cdot C \\ &= \bar{A}\bar{B} + (\bar{A}\bar{B} + AB) \cdot C \\ &= \bar{A}\bar{B} + \bar{A}BC + ABC \end{aligned}$$

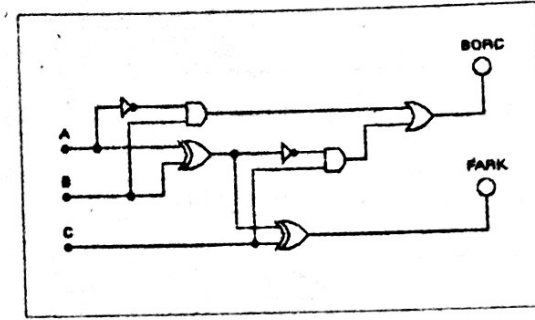
BC	00	01	11	10
A				
0		1		1
1	1		1	

$$FARK = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + ABC$$

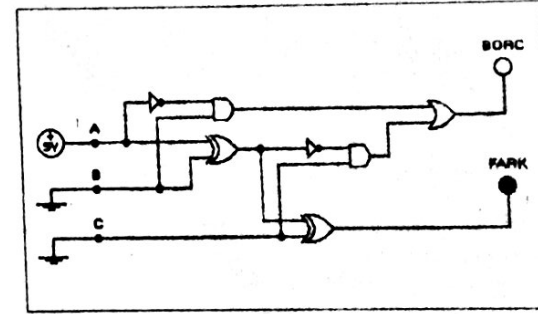
BC	00	01	11	10
A				
0		1	1	1
1			1	

$$BORÇ = \bar{A}\bar{B} + \bar{A}B + BC$$

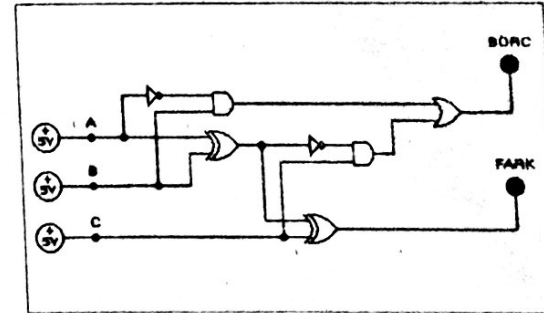
Bulunan FARK ve BORÇ ifadeleri, tasarım yoluyla bulunan çıkış ifadeleri ile aynı olduğundan, ikinci olarak çizilen tam çıkarıcı devresinin doğru olduğu sonucuna varılır.



Şekil 7.26: Tam çıkarıcı devresinin Electronics Workbench programında uygulanmış şekli



Şekil 7.27: A=1 ve B=C=0 durumunda tam çıkarıcının Fark çıkışının 1 olduğu durum



Şekil 7.28: A=B=C=1 durumunda Fark ve Borç çıkışlarının 1 olduğu durum

Tam çıkarıcı devresinin A girişi 1 potansiyelinde B ve C girişlerinin 0 olduğu durumda Fark çıkışının 1, BORÇ çıkışının ise 0 olduğu; tüm girişlerinin 1 olduğu durumda da her iki çıkışın 1 olduğu yapılan uygulamalar sonucu saptanmıştır.

3- 1'in Komplementine Göre Çıkarma:

İki tabanlı ve her ikisi de pozitif olan M-N çıkarma işlemi aşağıdaki gibi yapılır.

a) Çıkarılan M'i, çıkan N'in 1'in tümleyenine eklenir.

b) İlk adımda elde edilen sonuçta elde olup olmadığına bakılır.

- Elde varsa, LSD hanesine 1 eklenir.

- Elde yoksa, 1. adımda elde edilen sayının 1'e tümleyeni alınır ve önüne (-) işareti koyulur.

1'e Tümleyen: pozitif bir büyüklüğün her bitinin tersi alınarak oluşturulur.

Örnek 1: M = 1010100
N = 1000100

M-N işlemi yapınız.

N'in 1'e tümleyeni = 0111011

M ile N'in 1'e tümleyeni toplanır.

$$\begin{array}{r} 1010100 \\ + 0111011 \\ \hline 10001111 \\ + 1 \\ \hline 0010000 \end{array}$$

$$M-N = 10000$$

Örnek 2: M = 1000100
N = 1010100

M-N işlemi yapınız.

N'in 1'e tümleyeni = 0101011

$$\begin{array}{r} 1000100 \\ + 0101011 \\ \hline 1101111 \end{array}$$

elde yok

1101111 sayısının 1'e tümleyeni = 0010000

$$M-N = -10000$$

4) 2'nin Komplementine Göre Çıkarma:

Her ikisi de 2 tabanında olan iki pozitif sayının birbirinden çıkarılması (M-N) aşağıdaki gibi yapılır.

a) Çıkarılan M'i, çıkan N'in 2'ye tümleyenine ekleyin.

b) İlk adımda elde edilen sonuçta elde olup olmadığına bakılır.

- Elde varsa atılır.

- Elde yoksa, (a) adımında elde edilen sayının 2'ye tümleyeni alınır ve önüne bir (-) işareti koyulur.

2'ye Tümleyen: Sayının 1'e tümleyeni alınır ve bu sayının LSD'sinin altına 1 yazılarak toplanır.

Örnek 1: M = 1010100

N = 1000100

M-N işlemi yapınız.

N'in 2'ye tümleyeni: 0111011

$$\begin{array}{r} 0111011 \\ + 1 \\ \hline 0111100 \end{array}$$

$$\begin{array}{r} 1010100 \\ + 0111100 \\ \hline 10010000 \end{array}$$

son elde

$$M-N = 10000$$

Örnek 2: M = 1000100

N = 1010100

M-N işlemi yapınız.

N'in 2'ye tümleyeni: 0101011

$$\begin{array}{r} 0101011 \\ + 1 \\ \hline 0101100 \end{array}$$

$$\begin{array}{r} 1000100 \\ + 0101100 \\ \hline 1110000 \end{array}$$

elde yok

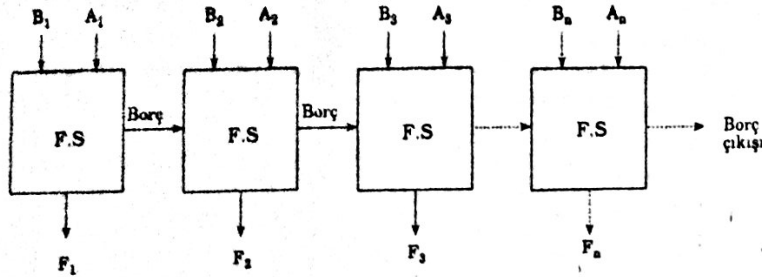
111000 sayısının 2'ye tümleyeni;

$$\begin{array}{r} 000111 \\ + 1 \\ \hline 001000 \end{array}$$

$$M-N = -1000$$

5) Paralel Çıkarcı:

n bitlik iki binary sayıyı çıkaran devrelerdir. Paralel toplayıcılarda olduğu gibi n sayıda tam çıkarcı (full subtractor) devresi kullanılır. Paralel çıkarcılarda, full subtractor'lar paket olarak gösterilir.



Şekil 7.29: Paralel çıkarcı devresinin blok diyagramı

Blok diyagramdaki borç çıkışı "1" ise çıkarmanın sonucu pozitif, "0" ise negatif olur.

4 bitlik toplama işleminin gerçekleştirildiği 7483 entegresi ile aynı zamanda 4 bitlik çıkarma işlemi de gerçekleştirilir. $C_0 = 0$ olduğu zaman toplama, $C_0 = 1$ olduğu sürece de çıkarma işlemi yapar. A bitleri entegreye aynen uygulanırken, B bitleri de değilleri alınarak entegreye uygulanır.

C) Çarpma Devresi

Binary sayılarda çarpma işleminin nasıl olduğu "Binary Sayılarda Dört İşlem" konusunda işlenmişti. Aşağıdaki örneklerde çarpma işlemi gerçekleştirilen lojik devrelere yer verilmiştir. Aynı zamanda çarpma işlemi tekrarlanan toplama işlemleri ile de gerçekleştirilebilir. 7485 entegresinin kaskad bağlanması ile çarpma devresi oluşturulabilir.

1) Çarpma Devresi Tasarımı ve Binary Çarpma Devresi

Örnek 1: 2 bitlik binary sayısının karesini alan lojik devreyi tasarlayınız.

GİRİŞLER		ÇIKIŞLAR			
A	B	F_3	F_2	F_1	F_0
0	0	0	0	0	0
0	1	0	0	0	1
1	0	0	1	0	0
1	1	1	0	0	1

Tablo 7.5: 2 bitlik binary sayısının karesini alan lojik devrenin doğruluk tablosu.

Lojik devrenin iki girişi vardır. $2^2 = 4$ değişik durumu bulunur. Önemli olan bu lojik devrenin kaç çıkış değişkeninin olmasıdır. Bunu bulmanın en basit yolu şudur: Giriş değişkenlerinin en büyüğü alınır. Bu örnekte girişin en büyük değeri decimal 3 sayısıdır $(11)_2$. Bunun karesi olan 9 sayısı en az 4 bit olarak yazılacağından bu devrenin çıkışı 4 tane-dir. Çıkış fonksiyonu da doğruluk tablosunda çıkışın her bir çıkış değişkeni için 1 olan yerlerin ifadesi yazılır.

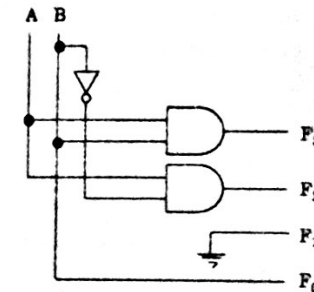
$$F_3 = AB$$

$$F_2 = A\bar{B}$$

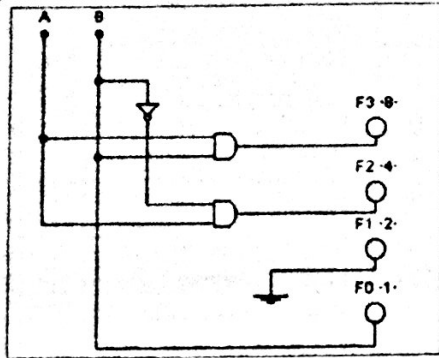
$$F_1 = 0$$

$$F_0 = \bar{A}B + AB = B(\bar{A} + A) = B$$

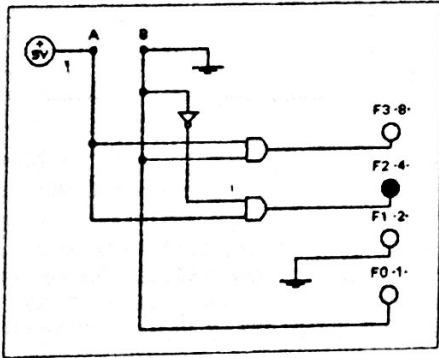
Bu çıkışlardan sadece F_0 çıkış değişkeni sadeleştirildi için bu devrenin en sade lojik diyagramı Şekil 7.30'daki gibi olur:



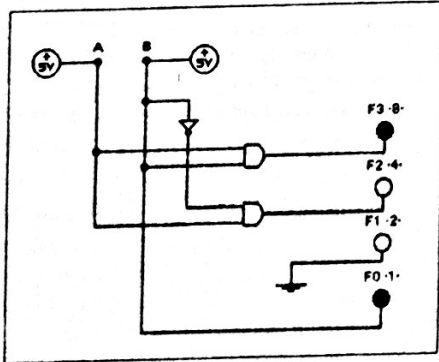
Şekil 7.30: 2 bitlik binary sayının karesini alan lojik devre



Şekil 7.31: Örnek 1'in Electronics Workbench programında uygulanmış şekli



Şekil 7.32: 10 sayısının karesi olan 0100 sayısının elde edilmesi



Şekil 7.33: 11 sayısının karesi olan 1001 sayısının elde edilmesi

A girişine 1, B girişine 0 uygulandığında bu devre 10 sayısının karesi olarak çıkışta 0100 sayısını vermiştir. Yine aynı devrenin A girişine 1, B girişine de 1 uygulandığında çıkıştan 1001 sayısı elde edilmiştir. Birinci uygulamada 2'nin karesi 4, ikinci uygulama da ise 3 sayısının karesi olan 9 sayısı elde edilmiştir.

Örnek 2: 3 bitlik binary sayının karesini alan lojik devreyi tasarlayınız.

GİRİŞLER			ÇIKIŞLAR					
A	B	C	(32) F ₅	(16) F ₄	(8) F ₃	(4) F ₂	(2) F ₁	(1) F ₀
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

Tablo 7.6: 3 bitlik binary sayının karesini alan lojik devrenin doğruluk tablosu

Bu lojik devrenin çıkışı 6 bit olmalıdır. Çünkü giriş değişkenlerinden en büyük sayı olan $(7)_{10} = (111)_2$ sayısının karesi olan $(49)_{10}$ sayısı binary olarak en az 6 bit olarak yazılabilir.

$$F_5 = \overline{A} \overline{B} C + \overline{A} B \overline{C} + A \overline{B} \overline{C} = \overline{A} B$$

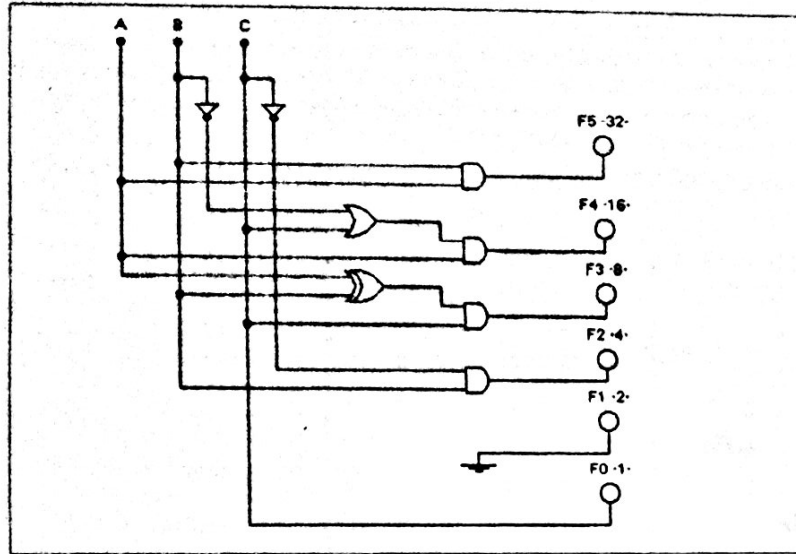
$$F_4 = \overline{A} \overline{B} C + \overline{A} B \overline{C} + \overline{A} B C = \overline{A} (B + C)$$

$$F_3 = \overline{A} B C + \overline{A} B \overline{C} = \overline{A} B (C + \overline{C}) = \overline{A} B$$

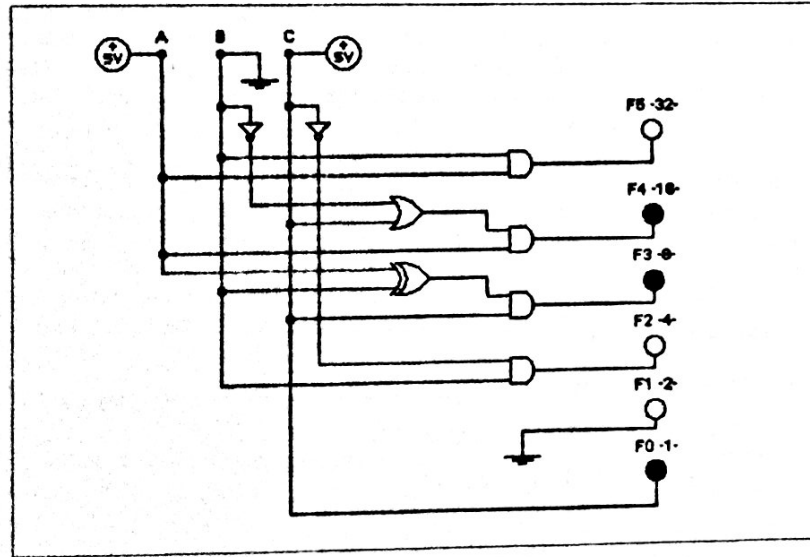
$$F_2 = \overline{A} B C + \overline{A} B \overline{C} = \overline{A} B (C + \overline{C}) = \overline{A} B$$

$$F_1 = 0$$

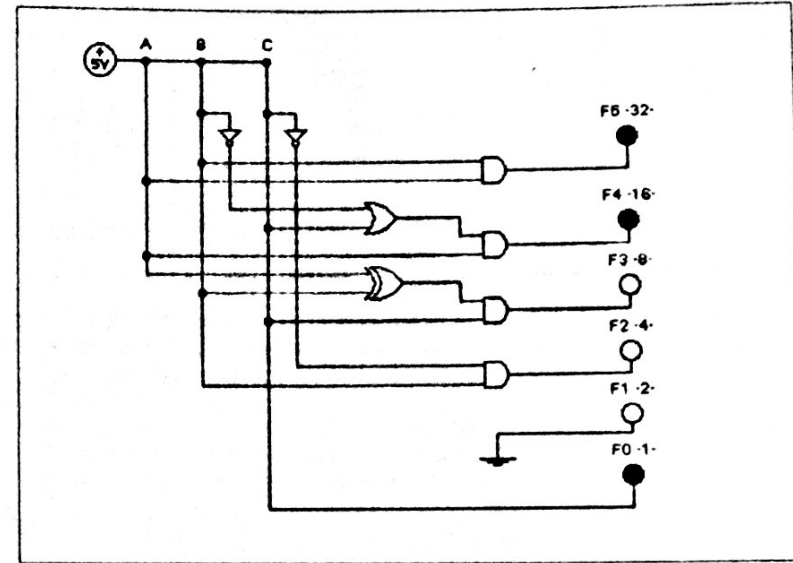
$$F_0 = \overline{A} \overline{B} C + \overline{A} B \overline{C} + \overline{A} B C + A \overline{B} \overline{C} = C$$



Şekil 7.34: Örnek 2'nin Electronics Workbench programında uygulanmış şekli



Şekil 7.35: 101 sayısının karesi olan 011001 sayısının elde edilmesi

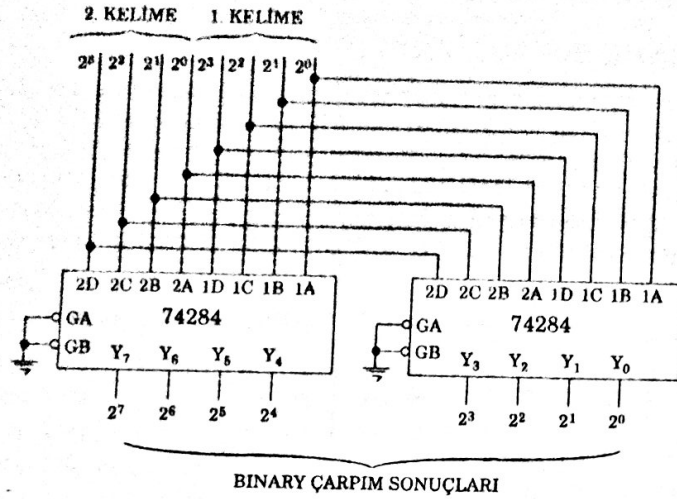


Şekil 7.36: 111 sayısının karesi olan 110001 sayısının elde edilmesi

Devrenin girişine 101 sayısı uygulandığında çıkıştan 011001 sayısı elde edilir. Çünkü 5 sayısının karesi 25'tir. Başka uygulama olarak devrenin girişine 111 uygulandığında çıkıştan 110001 sayısı alınır. 7 sayısının karesi olan 49 sayısı elde edilmiştir.

2- Çarpma İşlemi Yapan Entegre Devreler:

En hızlı çarpma işlemi gerçekleştiren devreler paralel binary çarpma üniteleridir. İki entegre devre kullanılarak 4 bitlik iki datanın çarpımını gerçekleştiren ve 8 bitlik sonucu 40 ns'de alan devrenin lojik diyagramı Şekil 7.37'de gösterilmiştir.



Şekil 7.37: 4 bitlik iki binary sayıyı çarpan lojik devre

D) KARŞILAŞTIRICILAR (COMPARATORS)

1- Karşılaştırıcı Devreler

Karşılaştırıcı devreleri; A ve B gibi iki sayıyı karşılaştırıp, $A < B$, $A = B$ ve $A > B$ durumunu üç binary değişkenle belirleyen devrelerdir. Aritmetik lojik devrelerinde önemli bir rol oynar. Böyle devreler toplama ve çıkartmayı yapmak için gereklidir. Karşılaştırıcı devreler, girişleri aynı veya farklı iken çıkış veren ve başka bir sistemi kontrol eden devrelerde, binary adres karşılaştırması yapan devrelerde de kullanılabilir. En basit karşılaştırma işlemi yapan kapı EX-OR kapısıdır. EX-OR kapısında girişler aynı iken (herikisi 0 veya 1) çıkış 0, girişler farklı iken (biri 0, diğeri 1) çıkış 1 dir.

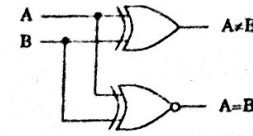
Örnek 1: Girişindeki iki bitü $A \neq B$ ve $A = B$ olarak ayırdeden devreyi tasarlayınız.

GİRİŞLER		ÇIKIŞLAR	
A	B	$A \neq B$	$A = B$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

$$A \neq B \text{ Çıkışı} = \bar{A}B + A\bar{B} = A \oplus B$$

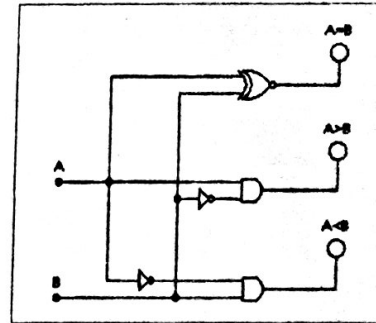
$$A = B \text{ Çıkışı} = \bar{A}\bar{B} + AB = A \odot B$$

Tablo 7.7: Basit karşılaştırıcının doğruluk tablosu

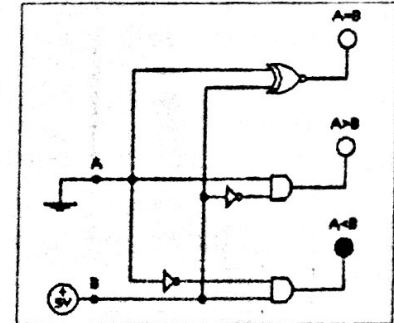


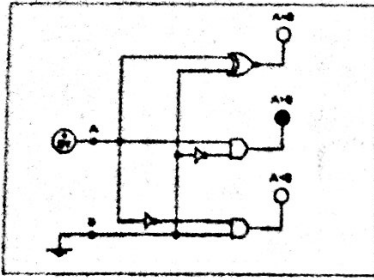
Şekil 7.38: Karşılaştırıcı devresi

$A=B=0$ olursa $A \neq B$ çıkışı "0", $A=B$ çıkışı 1 olacaktır. Girişlerden birisinin "1" olması durumunda (diğer giriş 0) $A \neq B$ çıkışı 1, $A=B$ çıkışı "0" olur. $A=B=1$ olduğunda ise $A \neq B$ çıkışı "0", $A=B$ çıkışı "1" olur. Bu devre ile girişindeki 0 ve 1 den oluşan sayıları basit olarak karşılaştırmış olduk.

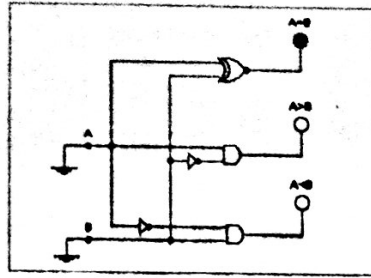


Şekil 7.39: İki bitlik karşılaştırıcı devresinin Electronics Workbench programında uygulanmış şekli

Şekil 7.40: A=0 ve B=1 durumunda $A < B$ çıkışının alınması



Şekil 7.41: A=1 ve B=0 durumunda A>B çıkışının alınması



Şekil 7.42: A=B=0 konumunda A=B çıkışının alınması

İki bitlik karşılaştırıcı devresini Electronics Workbench programında çalıştırdığımızda; girişlere aynı değerler uygulandığında A=B çıkışındaki lambanın, A=0 ve B=1 uygulandığında A<B çıkışındaki lambanın, A=1 ve B=0 uygulandığında A>B çıkışındaki lambanın ışık verdiğini gözleyebiliriz.

Örnek 2: Girişindeki iki biti A>B, A=B ve A<B olarak karşılaştıran lojik devreyi tasarlayınız.

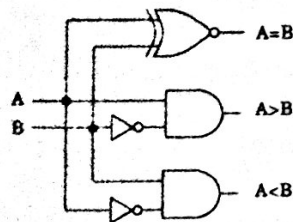
Girişler		Çıkışlar		
A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

$$A>B \text{ Çıkışı} = \overline{A}B$$

$$A=B \text{ Çıkışı} = \overline{A}\overline{B} + AB = A \odot B$$

$$A<B \text{ Çıkışı} = A\overline{B}$$

Tablo 7.8: İki bitli karşılaştıran devrenin doğruluk tablosu

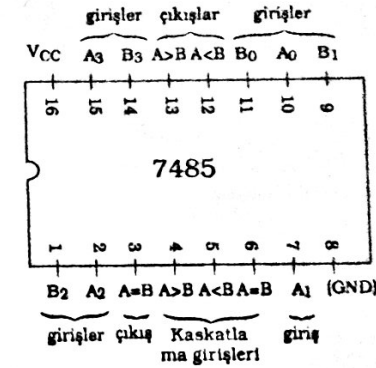


Şekil 7.43: İki bitli karşılaştıran devre

2) Entegre Devre Karşılaştırmacılar:

a) 7485 Entegresi:

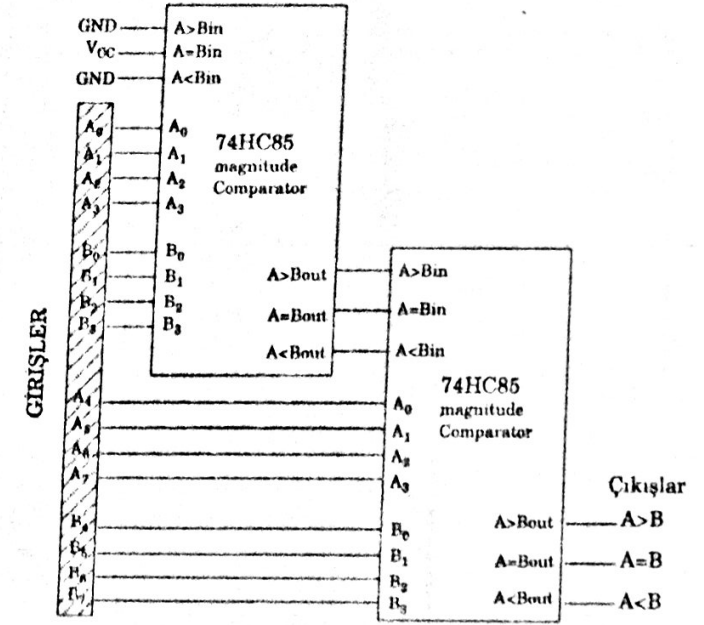
Bu entegre 4 bitlik iki binary veya BCD kodu karşılaştırma işleminde kullanılır. Karşılaştırılacak sayılar 4 bittten fazla ise (örneğin 7 bit ise) iki adet 7485 kullanılır. İlk 4 bit birinci 7485 entegresinin girişine, geri kalan 3 bit ise ikinci 7485'in A₀, A₁, A₂ ve B₀, B₁, B₂ uçlarına bağlanır. Ayrıca birinci 7485'in A>B, A=B ve A<B çıkışları ikincinin girişlerine bağlanır. İkinci 7485'in A>B, A=B ve A<B çıkışlarından devre çıkışları alınır.



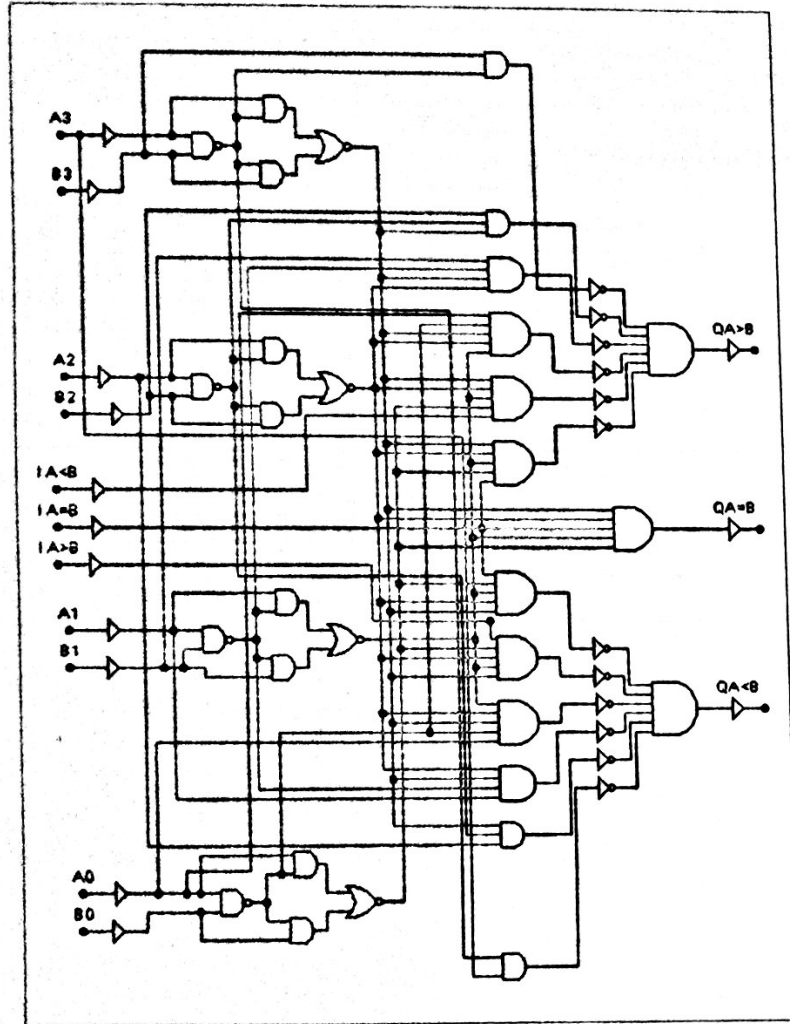
Şekil 7.44: IC 7485'in pin bağlantısı

Karşılaştırılan Girişler				Kaskatlama Girişleri			Çıkışlar		
A_3, B_3	A_2, B_2	A_1, B_1	A_0, B_0	$A > B$	$A < B$	$A = B$	$A > B$	$A < B$	$A = B$
$A_3 > B_3$	X	X	X	X	X	X	1	0	0
$A_3 < B_3$	X	X	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	1	0	0
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	X	X	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	0	0	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	0	1	1	0

Tablo 7.9: IC 7485'in fonksiyon tablosu



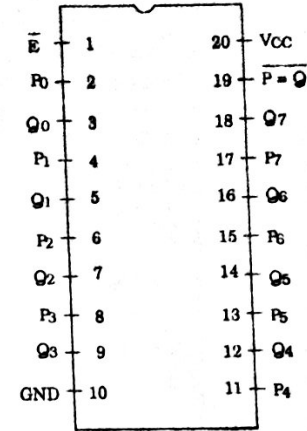
Şekil 7.45: Kaskad bağli genlik karşılaştırıcıları



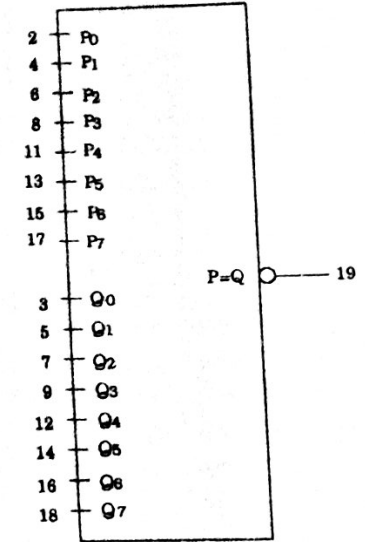
Şekil 7.46: 7483'ün Lojik diyagramı

b) 74688 Entegresi:

Bu entegre iki adet 8 bitlik karakteri karşılaştırır. Kategori olarak MSI sınıfına girer. 5 Volt pozitif kaynak voltajı ile çalışır.



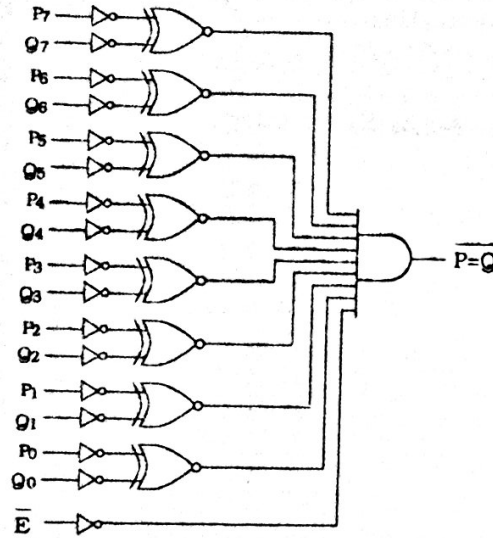
Şekil 7.47: Pin Bağlantısı



Şekil 7.48: Lojik Sembolü

Girişler		Çıkışlar
DATA P_n, Q_n	ENABLE $\overline{E}$	$\overline{P=Q}$
$P=Q$	0	0
x	1	1
$P>Q$	0	1
$P<Q$	0	1

Tablo 7.10: Fonksiyon Tablosu

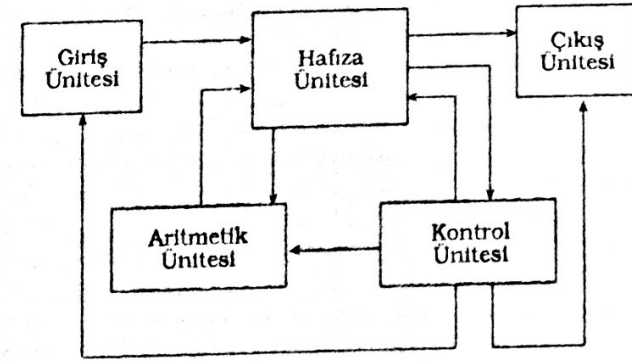


Şekil 7.49: IC 74688'in Lojik Diyagramı

E) ARİTMETİK LOJİK ÜNİTE (ALU)

Dijital bilgisayar, tanımlanan iki seviye arasındaki çalışmayı elektronik devrelerde toplayan bir cihazdır. İki seviyeli bir sistemin çalışma durumunda olduğu gibi bilgisayar devrelerinin çalışması da Boolean Cebri kanunlarına uyar. Bu bilgisayar çalışmalarına bilgi transferi, önerme ve hesaplama uygulanır. Bilgisayar tarafından yapılan aritmetik işlemler şüphesiz iki durumlu devreleri kullanmaya uygun olan ikili sistem esasına dayanır.

Dijital bilgisayar, genellikle Şekil 7.50'deki blok şema ile gösterilir.



Şekil 7.50: Sayısal (Dijital) bilgisayarın genel blok şeması

a) Hafıza Ünitesi

Hafıza ünitesinde "kelimeleri" depolayan "adreslere" sahip olan çok sayıda hücreler (cells) vardır. İkili bir sayı olan kelime bir program önermesi veya bilgi olabilir. Kontrol ünitesi aldığı önermeyi dekodlar (şifreyi çözer). Dekoder (kod, şifre çözücü) yapılması gerekli olan işlemin ne olduğunu ve çalışmayı yapmak için nasıl bilgi verebileceğini sisteme bildirir. Bilgisayarın kendi kendine işlem yapması, hafızadan bilgi alan ve kontrol ünitesine doğrudan doğruya önermede bulunan (kontrol ünitesine yapması gereken işlem hakkında önermede bulunan) aritmetik ünite (işlem ünitesi) yardımıyla gerçekleştirilir. Önermenin tamamlanmasında kontrol ünitesi hafıza tesbitindeki adrese göre aritmetik ünitenin yapacağı işlemi gösterir, hafızada tesbit edilen adresi depolar.

Kontrol ünitesi, hafızada toplanan ek bilgiyi de transfer ederek giriş ünitesine verir. Aynı şekilde diğer görevi de, sonucu yayınlamak için bilgiyi çıkış ünitesine vermektir.

b) Giriş Ünitesi

Bu ünite, teletype (telli otomatik daktilo), (anahtara basıldığında her seferinde birçok sayıda pals üreten cihaz), delinmiş bant, basılmış bantlar, manyetik bant (bilgi paketleme yoğunluğu çok yüksek seviyede olan bant) ve manyetik silindirik gibi ünitelerden biri olabilir.

Giriş bilgisi besleme metodu, gerekli bilgi oranına göre seçilir.

c) Çıkış Ünitesi

Bu ünite kullanılan başlıca eleman, bir gösterge elemanıdır. Böyle bir ünite, özel yöntemler kullanılarak saniyede yüzlerce sembolü gösterme kabiliyetine sahiptir. Çıkış ünitesinin diğer tipleri: Grafik ve katot ışınlu tüpü olan displaylerdir.

d) Hafıza Ünitesi

Computerdeki hafıza hücreleri manyetik TOROID'ler, elektronik flip floplar ve manyetik silindirdir veya bant da olabilir. Bu değişik tipler arasındaki başlıca fark ayrı ayrı hafıza hücrelerinin bulunmasıdır. Örneğin manyetik toroitler veya flip-flop'lar direkt olarak ve her hücreye hemen geçiş sağlarken, çeşitli manyetik bant tipleri gerekli bilgi için kaynağa ihtiyaç gösterir. Bilgi okumak için, manyetik bant başa kadar getirilmelidir. Açık bir şekilde ifade etmek gerekirse, manyetik bant sistemi çok daha yavaştır. Fakat diğer taraftan çok büyük bir hafızanın oluşturulmasına imkan verir.

e) Kontrol Ünitesi

Bu ünite, fonksiyonun önerme kodlamasını, lojik kararlarını ve tam sistem zamanlamasını yapmaya yarayan çok sayıdaki lojik devrelerden oluşur.

f) Aritmetik Ünitesi

Bu ünite hesaplama ünitesidir. Burada toplama, çıkarma, çarpma ve bölme işlemleri yapılır. Basit işlemlerin programlanmış serilerinden yapılmış daha karmaşık işlemler yine burada sonuçlandırılır. Programlama, kontrol ünitesi tarafından yapılmak üzere hafızaya verilen önermelerin işleme konulmasıdır.

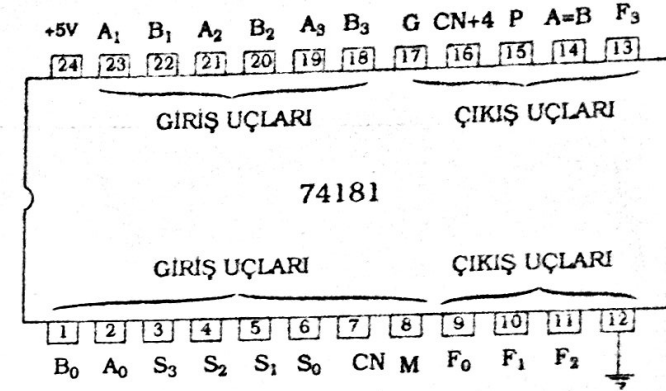
1- 74181 ALU Entegresinin Özellikleri

74181 Entegresi, çeşitli kapı devreleri ile aritmetik devrelerin yaptığı işlemleri tam olarak yerine getirebildiği gibi bazı ek işlemleri yapabilecek durumdadır. Bu entegrenin yaptığı işlemler iki ana grupta toplanır:

- Aritmetik işlemler (toplama, çıkarma, çarpma gibi)
- Lojik işlemler (AND, OR, NAND, EXOR gibi)

24 bacaklı olan bu devrenin uçlarından biri, söz konusu ana grup-

lardan birini seçmede kullanılır. Bu uç M ile gösterilir ve devrenin 8 no-lu ucudur. Devre, M=0 olduğunda aritmetik işleme, M=1 olduğunda ise lojik işleme programlanmış demektir.



Şekil 7.51: 74181 Entegresinin Bacak Tanımlaması

Diğer uçlardan dört tanesi ikili kod formunda ifade edilmiş A kelimesi girişi için, 4 tanesi B kelimesi girişi için, 4 tanesi de A ve B kelimeleri arasında yapılan aritmetik veya lojik işlemleri sonucu ortaya çıkan F kelimesi çıkışı içindir.

Bacaklardan bir diğer dört tanesi de yapılacak aritmetik veya mantık işlemini seçmede kullanılır. Bu uçlar s_0, s_1, s_2, s_3 olarak adlandırılmıştır.

74181 entegresini kullanırken, yapılacak işlemi tanımlamak için önce M ucunu programlayıp ana işlem grubunu belirlemek (aritmetik veya lojik olup olmadığı) sonra da, s_0, s_1, s_2, s_3 dizisi ile ana grup içinde yapılacak işlemi (toplama, çıkarma, AND, OR gibi) tanımlamak gerekir. Bunlar Tablo 7.10'da verilmiştir.

2- Fonksiyon Tablosu:

SEÇME TABLOSU s ₃ s ₂ s ₁ s ₀	M = 1 Lojik İşlemler	M=0. ARİTMETİK İŞLEMLER	
		C _n = 1	C _n = 0
0 0 0 0	F = $\bar{A}$	F = A	F = A artı 1
0 0 0 1	F = $A + \bar{B}$	F = A + B	F = (A+B) artı 1
0 0 1 0	F = $\bar{A}B$	F = A + $\bar{B}$	F = (A+B) artı 1
0 0 1 1	F = 0	F = eksi 1	F = sıfır
0 1 0 0	F = $\bar{A}\bar{B}$	F = A artı $\bar{A}\bar{B}$	F = A artı $\bar{A}\bar{B}$ artı 1
0 1 0 1	F = $\bar{B}$	F = (A+B) artı $\bar{A}\bar{B}$	F = (A+B) artı $\bar{A}\bar{B}$ artı 1
0 1 1 0	F = $A \oplus B$	F = A eksi B eksi 1	F = A eksi B
0 1 1 1	F = $A\bar{B}$	F = $A\bar{B}$ eksi 1	F = $A\bar{B}$
1 0 0 0	F = $\bar{A} + B$	F = A artı AB	F = A artı AB artı 1
1 0 0 1	F = $\bar{A} \oplus B$	F = A artı B	F = A artı B artı 1
1 0 1 0	F = B	F = (A+B) artı AB	F = (A+B) artı AB artı 1
1 0 1 1	F = AB	F = AB eksi 1	F = AB
1 1 0 0	F = 1	F = A artı A	F = A artı A artı 1
1 1 0 1	F = $A + \bar{B}$	F = (A+B) artı A	F = (A+B) artı A artı 1
1 1 1 0	F = A + B	F = (A+B) artı A	F = (A+B) artı A artı 1
1 1 1 1	F = A	F = A eksi 1	F = A

Tablo 7.11: Pozitif lojiğe göre (1 = +5V, 0 = 0V) 74181 Entegresinin Fonksiyon Tablosu

Lojik işlemlerinde (+) işareti kullanıldığı için, herhangi bir karışıklığa yol açmamak için, aritmetik işlemlerde (+), (-) işaretleri yerine "artı", "eksi" ifadeleri kullanılmıştır.

Örneğin, 74181 entegresi ile iki sayı toplanmak istenildiğinde,

- 1) Sayılardan biri binary formda A₃, A₂, A₁, A₀ uçlarına diğeri de B₃, B₂, B₁, B₀ uçlarına uygulanır.
- 2) Yapılacak işlem aritmetik olduğundan M=0 yapılır.
- 3) Devreyi toplama işlemine programlamak için s₃s₂s₁s₀=1001 yapılır.

74181'in uçlarından iki tanesi besleme gerilimi için, bir tanesi C_n (elde var) çıkışı için, bir tanesi C_n + 4 (elde var) çıkışı için ve geri kalan üç tanesi de çeşitli özel amaçlar için kullanılır.

Tablo 7.11'den görüldüğü gibi 74181 devresini 5 tip aritmetik işle-

me, 16 tip lojik işleme ve 8 tip hem lojik hem aritmetik işleme programlamak mümkündür. 3 adet mantık işlemi tekrarlardır. Böylece, 74181 entegresi ile 32 çeşit işlem yapma imkanı vardır.

74181 entegresinde aritmetik işlemlerde decimal yerine hexadecimal sistem kullanılır. Örneğin 2+11 işleminde, decimal sistemde, yedili göstergede 3 rakamı, elde var çıkışında "1" belirlir. Halbuki hex sistemde; 7 li göstergede 13 sayısını ifade eden $\square$ işareti oluşur, elde var çıkışında da "0" vardır. Diğer örnek olarak 8 ile 11 toplandığında;

Decimal sistemde; elde var ve 9.

Hexadecimal sistemde; elde var ve 3 oluşur.

Decimal	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
Hex	0	1	2	3	4	5	6	7	8	9	c	d	e	f		

Tablo 7.12: Yedili göstergede decimal ve hexadecimal sisteme göre oluşacak şekiller

Decimal sistemde 9'dan sonra 0'a dönüldüğü halde (elde var 1 olarak), hexadecimal sistemde 15'den sonra 0'a dönülmektedir. Bu durumda bir 74181 entegresi ile, en çok 15+16 = 31'e kadar toplama yapmak mümkündür.

74181 entegresi lojik işlemi yaparken A ve B kelimelerinin ilgili bitlerini ayrı ayrı işleme sokar. Örneğin, 74181 entegresi F=AB lojik işlemine programlandığında,

$$F_0 = A_0 \cdot B_0$$

$$F_1 = A_1 \cdot B_1$$

$$F_2 = A_2 \cdot B_2$$

$$F_3 = A_3 \cdot B_3 \text{ olarak 4 AND işlemini ayrı ayrı yapar.}$$

3- Aritmetik İşlemci Devresi:

3 ve 8 rakamlarını 74181 entegresi ile toplamak istediğimizde;

$$C_n (7 \text{ nolu uç}) = 1$$

M (8 nolu uç) = 0

$s_3 s_2 s_1 s_0$ (3, 4, 5 ve 6 nolu uçlar) = 1001

$A_3 A_2 A_1 A_0$ (19, 21, 23 ve 2 nolu uçlar) = 0011

$B_3 B_2 B_1 B_0$ (18, 20, 22 ve 1 nolu uçlar) = 1000 uygulanır.

Sonuçta:

$F_3 F_2 F_1 F_0$ (13, 11, 10, 9 nolu uçlar) = 1100 ve $C_n + 4 = 1$ oluşur.

$A=9$ dan $B=5$ çıkarılmak istendiğinde;

C_n (7 nolu uç) = 0

M (8 nolu uç) = 0

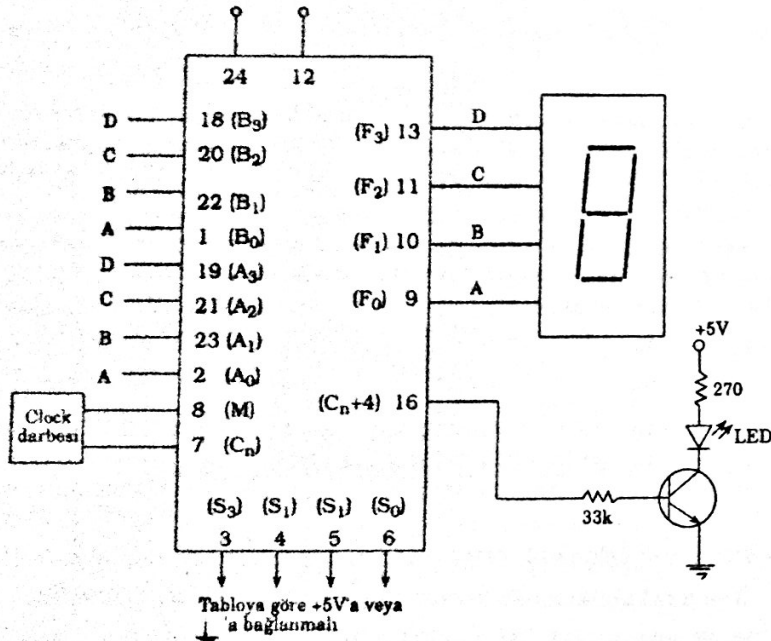
$s_3 s_2 s_1 s_0$ (3, 4, 5, 6 nolu uçlar) = 0110

$A_3 A_2 A_1 A_0$ (19, 21, 23, 2 nolu uçlar) = 1001

$B_3 B_2 B_1 B_0$ (18, 20, 22, 1 nolu uçlar) = 0101 uygulanır. Sonuçta;

$F_3 F_2 F_1 F_0$ (13, 11, 10, 9 nolu uçlar) = 0100

$C_n + 4$ (16 nolu uç) = 1 oluşur.



Şekil 7.52: Aritmetik İşlemci Devresi

Örnek: $A_3 A_2 A_1 A_0 = 0000$

$B_3 B_2 B_1 B_0 = 0000$

$s_3 s_2 s_1 s_0 = 1001$

$M = 0$

$C_n = 1$ yapınız ve aşağıdaki tablonun yedili gösterge ve elde var kısmını doldurunuz.

A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	Yedili gösterge	Elde var
0	0	0	0	0	0	0	0	0	1
0	1	0	1	0	0	1	1	8	1
0	1	0	1	0	1	1	1	11	1
1	0	0	0	1	0	0	0	0	0
1	0	1	0	1	1	0	0	6	0
1	1	1	1	1	1	1	1	15	0

Örnek: $C_n = 0$

$s_3 s_2 s_1 s_0 = 1110$ yaparak entegreyi çıkarma yapmak için programlayınız. Başlangıçta $A_3 A_2 A_1 A_0 = 0000$, $B_3 B_2 B_1 B_0 = 0000$ alınız. Aşağıdaki tablonun yedili gösterge ve borç hanelerini doldurunuz.

A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	Yedili gösterge	Borç
0	1			0	0	0	1	3	0
1	0			0	1	0	1	3	0
1	1			0	0	1	1	11	0
0	0			0	1	0	0	1	1
0	0			0	1	1	1	9	1
0	0			1	1	1	1	1	1

ÜNİTE VII İLE İLGİLİ SORULAR

- 1) Yarım çıkarıcı devresinin tablosunu yapınız. Fonksiyonunu çıkararak sadece NAND kapıları ile gerçekleştiriniz.
- 2) 3 bitlik iki sayıyı toplayabilen bir paralel toplayıcı devresinin blok diyagramını çizerek $(5+7)$ işlemini devre üzerinde yapınız.
- 3) 4 bitlik paralel toplayıcı çizerek $(3+5)$ işlemini yapınız.